

CAPÍTULO 1

Introducción.

1.1. Automatización.

Automatización es un proceso que se ha convertido en una necesidad actual para las industrias. Quienes no la utilizan como una característica de su infraestructura, quedan por completo fuera de la competencia. Es así, como automatizar se convierte en la acción fundamental de una nueva Revolución Industrial.

La automatización es el resultado de transferir años de investigación científica y tecnológica a nuevas aplicaciones prácticas. En esta transición, una de las áreas que mayor impacto ha tenido es la robótica. Los nuevos sistemas de manufactura están compuestos en su mayoría por robots industriales.

1.2. Robots Industriales.

Los robots industriales no se ven como humanos aunque hacen el trabajo de los humanos. (Koren, 1985). Es aquí donde la definición que describe a un robot como un aparato mecánico que se parece y hace el trabajo del hombre, pierde casi por completo su validez. Entonces el concepto de un robot se complementa y se considera como un manipulador reprogramable y multifuncional concebido para llevar a cabo tareas diversas (*Robot Institute of America*, Koren 1985).

Los robots manipuladores actuales son los descendientes directos de las máquinas de control numérico (NC) (Endelberg en Zomaya, 1994).

Las primeras máquinas automáticas fueron diseñadas para realizar tareas específicas, si surgían modificaciones en la producción, estas máquinas tenían que renovarse. En 1954, George C. Devol, patentó la idea de un aparato mecánico que fuera capaz de programarse las veces que fuera necesario para realizar tareas requeridas por quienes lo adquirieran (Pawson, 1985). El primer robot industrial fue instalado en *Unimation Inc.* en 1961. A partir de entonces, se inició una implantación masiva de robots en industrias alrededor del mundo.

La estructura típica de un robot industrial (Figura 1.1) consiste básicamente en:

┌ Manipulador.

Es la estructura mecánica del robot. Un conjunto de barras conectadas por pares cinemáticos de modo que constituyan una cadena cinemática abierta.

┌ Fuente de poder.

Proporciona la Energía necesaria a los actuadores del manipulador.

┌ Controlador.

Sus funciones básicas son:

- ◆ La iniciación y finalización del movimiento de los componentes individuales del manipulador en una secuencia de puntos especificados.
- ◆ Almacenamiento en su memoria de datos acerca de la posición y secuencia de movimientos.
- ◆ Permite al robot interactuar con el entorno por medio de sensores.

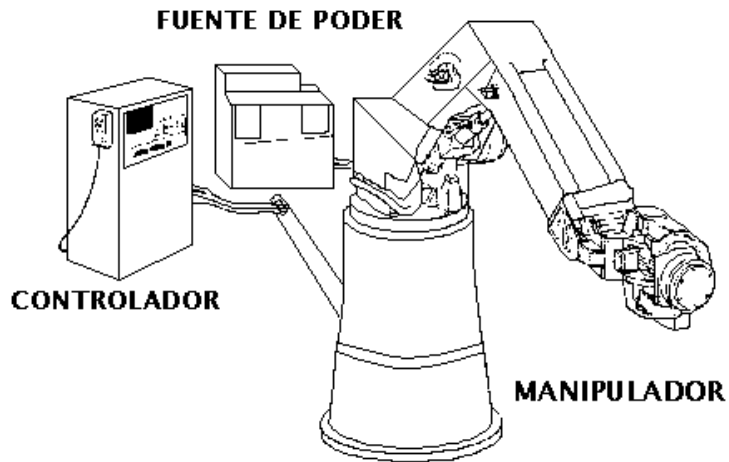


Figura 1.1 Estructura típica de un robot industrial.

Hoy en día, las computadoras juegan un papel importante en el control de los robots, haciéndolos superiores a otras máquinas que usualmente aparecen en los pisos de las fábricas. El uso extensivo de los robots industriales, responde a la creciente demanda de automatización flexible (Zomaya, 1994).

Existen muchas aplicaciones de robots industriales, sin embargo entre los primeros usos que se les dieron y que siguen siendo los más utilizados destacan: la soldadura por puntos y la pintura de carrocerías de automóviles. En este proyecto, se atacarán algunos aspectos de problemas análogos al primer caso, la soldadura de puntos.

1.3. Problemas en las aplicaciones industriales.

A pesar del amplio desarrollo que respalda la tecnología de los robots, aún encontramos algunos problemas que acompañan a las aplicaciones industriales, sin

embargo, existe investigación suficiente que puede ofrecer soluciones para dichos problemas.

En la actualidad, los robots industriales son controlados por métodos de tecnología básica y las aplicaciones tecnológicas están por debajo de los avances científicos (Overgaard, et al. 1998). Es decir, a pesar de que en el área de la robótica existen investigaciones sobre sensores, estrategias que detectan y evitan colisiones y métodos eficientes de planificación de trayectorias, en su mayoría no han sido implantados. Y es que por lo regular, la transición de la teoría a la práctica no siempre es fácil (Overgaard, et al. 1998).

1.4. Programación de los robots industriales.

La programación de los robots industriales se divide básicamente en dos métodos que se describen a continuación:

1.4.1. *On-line programming.*

Requiere de una interacción directa de la gente con el robot. Exige un amplio conocimiento de los comandos y de las acciones del autómeta, cualquier descuido puede causar graves daños en el equipo y en la seguridad del operador. Se divide a su vez en tres procedimientos:

┌ *Lead-through teaching.*

Es el método más simple. El operador debe llevar manualmente al robot a todos los puntos que alcanza durante la realización de alguna tarea. Las trayectorias y los puntos se almacenan en su memoria. Muchas veces se utilizan robots de tamaño y peso menores al real para facilitar su manejo al operador. Por lo regular, este método se utiliza en la pintura de carrocerías (Figura 1.2).

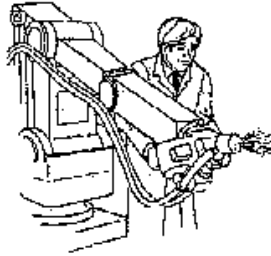


Figura 1.2 *Lead-through teaching*

┌ *Teaching pendant programming.*

Utiliza un control remoto para desplazar el brazo del robot a las diferentes posiciones. Su sistema, almacena únicamente los puntos terminales indicados por el operador y utiliza algoritmos que calculan los puntos intermedios en las trayectorias de uno a otro. Los tipos de desplazamiento pueden ser: lineales, circulares o de algún diseño establecido. Se utiliza en ambientes sofisticados, el brazo se maneja en coordenadas cartesianas. Al final, se ejecuta la secuencia de movimientos de punto a punto (Figura 1.3).

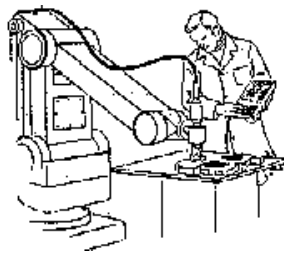


Figura 1.3 *Teaching pendant programming.*

└ *Textual programming.*

Utiliza la programación de un lenguaje de alto nivel, implica el uso del sistema de control del robot.

1.4.2. *Off-line programming.*

Propone el desarrollo de programas de control a través de una computadora distinta a la del sistema del robot. La idea ha ganado importancia con los paquetes de simulación CAD (*Computer Aided Design*).

Las ventajas que ofrece se resumen a las siguientes:

- └ Hace posible la programación del robot antes de instalar el sistema completo.
- └ Utiliza sistemas de uso común basados en computadoras personales.
- └ Elimina la necesidad de especializar programadores en el ambiente del robot.
- └ Reduce el tiempo requerido para crear un programa y realizar cambios sobre el mismo.
- └ Permite visualizar la ejecución de una tarea en la pantalla de una computadora eliminando distracciones graves de los operadores y sus consecuencias.

Entre sus desventajas, podemos mencionar las siguientes:

- └ Hay veces que surgen diferencias entre la tarea simulada y la operación real del robot, aunque este problema se debe principalmente a errores de calibración.
- └ No todos los robots tienen una arquitectura abierta por lo que no se pueden desarrollar paquetes que permitan la programación del tipo *Off-line programming*.
- └ La computadora donde se desarrolla el paquete requiere de una vía de comunicación con el robot.
- └ Se debe considerar la existencia de un amplio número de lenguajes de programación de los robots.

(Glagowski, T. et al., 1992)

De los métodos que fueron mencionados, los más utilizados en la industria son los de On-Line programming, principalmente, aquellos que emplean la programación manual. El problema fundamental de estas aplicaciones es que la detección de colisiones, la coordinación de los robots, la velocidad de ejecución y el orden de recorrido de un punto a otro quedan bajo el criterio y la responsabilidad del operador.

1.5 Un caso real: Volkswagen.

En nuestro país, una de las industrias con mayor automatización es la empresa Volkswagen. Esta empresa cuenta con aplicaciones del campo de la robótica dentro de su proceso de producción de automóviles y que van desde el transporte de piezas hasta la pintura de carrocerías.

Actualmente los robots industriales de la planta tienen un sistema de programación manual (Figura 1.4). Este sistema utiliza una interfaz gráfica con la que se puede lograr que los robots ejecuten tareas como soldadura y pintura de autos dentro de la línea de producción, entre otros.



Figura 1.4 Control KCP para la programación manual de los robots, Volkswagen.

El sistema de programación es de tipo *teaching pedant programming* y de manera general funciona de la siguiente forma:

Supongamos que queremos soldar una pieza en ocho puntos distintos, como se muestra en la Figura 1.5:



Figura 1.5 Puntos de soldadura en una pieza de carrocería.

Para que el robot sea capaz de realizar esta tarea es necesario llevarlo manualmente por medio de la interfaz de programación a cada punto (Figura 1.6).

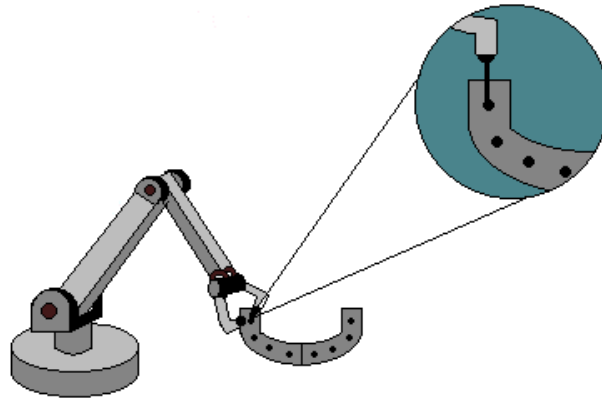


Figura 1.6 Ubicación del robot en un punto de soldadura.

El proceso comienza cuando se coloca al robot en una posición inicial y en seguida se graba esta configuración; posteriormente se debe colocar el elemento terminal del robot en el primer punto que se desea soldar y volver a guardar dicha posición, este proceso se repite todas las veces que sea necesario hasta terminar de recorrer los puntos que el robot debe soldar en la pieza.

Comment [UO1]:

Además de las configuraciones que alcanzan los puntos deseados también se deben indicar cuáles son las posiciones intermedias, es decir, las configuraciones que le permiten al robot separarse de la pieza para colocarse en otro punto (Figura 1.7).

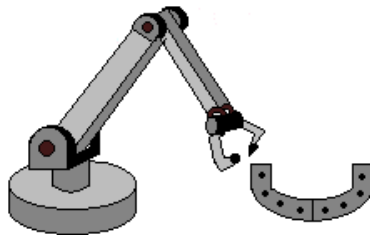


Figura 1.7 Puntos intermedios de soldadura.

Al terminar el proceso, se obtiene un programa que contiene un conjunto de configuraciones con las que el robot realiza la tarea de soldadura y se procede a ejecutar dentro de un ciclo. El camino entre dos configuraciones es desconocido por el programador y es determinado por el software de programación.

Este método es *On-Line programming*, como se mencionó anteriormente, es muy utilizado en las aplicaciones industriales de los robots sin embargo es un proceso de programación que puede durar varias horas o tal vez hasta algunos días, dependiendo de la cantidad de puntos que se desean alcanzar o bien de los cambios continuos de posiciones grabadas, que no fueron muy exactas en su colocación. A esto debemos agregar que si la pieza soldada se cambia por otra (por ejemplo, una más pequeña) o se modifica su posición o su orientación, puede causar en el peor de los casos, la reprogramación completa de los puntos.

Podemos observar que existe una necesidad por minimizar la distancia en las trayectorias de un punto a otro, mejorando su tiempo de recorrido y su productividad donde lo más importante es agilizar su proceso de programación.

Específicamente, el objetivo de la tesis consiste en generar un prototipo que facilite la programación de los robots y que proporcione un conjunto de trayectorias "óptimas" al visitar un número finito de puntos. Este proyecto está inspirado en un problema de un caso real de la planta de Volkswagen.

1.6 Objetivos del proyecto.

El sistema actual de programación de los robots en la planta es simple, principalmente por la interfaz gráfica que tiene y por el concepto que maneja, sin embargo presenta algunos problemas que podrían solucionarse, por ejemplo:

Tipos de desplazamiento bajo criterio del programador.

El sistema cuenta con tres tipos de desplazamiento para ir de un punto a otro, y que son llamados: punto a punto, lineal y circular. Estos desplazamientos se determinan al grabar la configuración de un punto específico y quedan, en cierto modo, al criterio del programador.

Exactitud para colocar al robot en un punto específico.

El control manual con que se mueve el robot de un punto a otro requiere que su programador adquiera conocimientos sobre los ejes de coordenadas del robot y además se necesita tiempo para que una persona sea capaz de dominarlo. Estos problemas parecen no tener importancia, sin embargo el control manual es utilizado para colocar al robot en un punto específico por lo tanto la exactitud para alcanzar un punto depende de las estimaciones del programador y afecta en la calidad del producto final.

Duración del tiempo de programación.

Otro problema que se puede observar es que el sistema emplea demasiado tiempo en su programación, pues se requiere de un largo proceso para que un robot realice alguna tarea con una pieza de tamaño y forma específicos. Como se mencionó, este proceso puede durar varias horas o inclusive algunos días.

Problemas al modificar la orientación y tamaño de la pieza.

Por último, si se tiene el problema de visitar los mismos puntos pero con un elemento terminal distinto (por ejemplo, el tamaño o longitud de una pinza), se requiere que el robot reconozca este nuevo elemento, lo que podría provocar que el robot tuviera que ser reprogramado.

En resumen, los problemas que se desean resolver con este proyecto se enfocan al tiempo de programación empleado, a la falta de optimización en las trayectorias y que tiene que ver con los distintos tipos (proporcionados por el sistema) de desplazamiento del robot de un punto a otro, además de evitar los errores humanos.

El criterio de optimización de trayectorias de un punto a otro es, evidentemente la distancia que existe entre las configuraciones de estos puntos.

El algoritmo desarrollado para la optimización de trayectorias, será probado por medio de un simulador en dos dimensiones que muestre el movimiento del robot con las trayectorias óptimas que resulten del algoritmo.